

CUESTIONARIO

El curso asume un conocimiento previo de programación y arquitectura de la plataforma Arduino. Contestar correctamente la mitad o más preguntas de este test es un buen indicador de cumplir con el nivel mínimo que asegura un aprovechamiento del curso.

NOTA: Para cada pregunta, sólo una de las opciones es correcta. Las soluciones correctas se indican en una tabla al final del test.

1	Arduino es:	
a	Una tarjeta de desarrollo.	
b	Una familia de tarjetas de desarrollo.	
c	Un compilador para microprocesadores.	
d	Un entorno de desarrollo.	
e	A+C	
f	A+D	
g	B+C	
h	B+D	
2	¿Qué es un escudo (shield)?	
a	Una tarjeta que se apila sobre una placa de desarrollo Arduino para ampliar sus características.	
b	Una tarjeta que protege contra sobretensiones.	
c	Una tarjeta que evita las interferencias radioeléctricas.	
d	Un sistema que se usa para encapsular las tarjetas de desarrollo.	
3	¿Cuál es la intensidad máxima que se puede obtener de un puerto digital?	
a	No se puede obtener intensidad.	
b	10 mA.	
c	25 mA.	
d	40 mA.	
e	50 mA.	
4	¿Qué protocolos de comunicación síncrona incorporan como función especial los terminales digitales de una tarjeta Arduino?	
a	SPI.	
b	I ² C.	
c	Ethernet.	
d	Todas las anteriores.	
e	A+B.	
f	B+C.	
g	A+C.	
h	Ninguna de las anteriores.	

5	Utilidad del "Serial monitor".	
a	Enviar y recibir datos textuales de la placa Arduino.	
b	Transmitir el sketch desde el PC al microcontrolador.	
c	La opción A pero sólo si la velocidad de transmisión seleccionada para "Serial monitor" es la misma que se haya especificado en el sketch ejecutado en la placa.	
d	La opción B pero sólo si la velocidad de transmisión seleccionada para "Serial monitor" es la misma que se haya especificado en el sketch ejecutado en la placa.	
e	A+B.	
f	C+D.	
6	El tipo de dato "int" en la tarjeta Arduino DUE puede almacenar:	
a	Un valor entero entre 0 y 4294967294	
b	Un valor entero entre 0 y 65536	
c	Un valor entero entre -32768 y 32767	
d	Un valor entero entre -2147483648 y 2147483647	
7	La resistencia PULL-UP:	
a	Se activa mediante la constante predefinida INPUT-PULLUP	
b	Se activa mediante la constante predefinida INPUT	
c	Se desactiva mediante la constante predefinida INPUT-PULLUP	
d	Se desactiva mediante la constante predefinida INPUT	
e	B+C	
f	A+D	
8	El operador módulo:	
a	Tiene como símbolo %	
b	Tiene como símbolo \$	
c	Devuelve el cociente de una división	
d	Devuelve el resto de una división	
e	A+C	
f	A+D	
g	B+C	
h	B+D	
9	¿Qué significa cambiar el tipo de una variable 'al vuelo'?	
a	No existe este concepto en Arduino	
b	Cambiar el tipo de la variable desde ese momento	
c	Intercambiar el tipo de dos variables	
d	Cambiar el tipo de la variable en esa línea del sketch	

10	¿Se puede acceder simultáneamente a los dos bloques de memoria en la arquitectura Harvard?	
a	Sí	
b	No	
c	Depende del tipo de tarjeta	
d	Depende del sistema operativo	
11	La memoria FLASH frente a la EEPROM	
a	Es más rápida	
b	Tiene mayor densidad	
c	Tolera menos ciclos de escritura/borrado	
d	Todas las anteriores	
e	A+B	
f	B+C	
g	A+C	
h	Ninguna de las anteriores	
12	El tamaño máximo de un programa de Arduino compilado está limitado:	
a	Por la capacidad de memoria FLASH	
b	Por la dimensión SRAM	
c	Por el tamaño de la memoria EEPROM	
d	Depende del tipo de tarjeta	
13	El espacio reservado para las variables del problema en un programa de Arduino compilado está limitado:	
a	Por la capacidad de memoria FLASH	
b	Por la dimensión SRAM	
c	Por el tamaño de la memoria EEPROM	
d	Depende del tipo de tarjeta	
14	La activación del objeto Serial	
a	Consiste en abrir el puerto serie	
b	Consiste en fijar su velocidad	
c	Las tareas a) y b) se realizan con funciones independientes	
d	Las tareas a) y b) se realizan con una misma función	
e	A+B+C	
f	A+B+D	
g	Ninguna de las anteriores	

15	Los métodos fundamentales que incorpora la librería oficial EEPROM del entorno IDE de Arduino son:	
a	Write(posición,valor)	
b	Print(posición, valor)	
c	Read(posición)	
d	Todas las anteriores	
e	A+B	
f	B+C	
g	A+C	
h	Ninguna de las anteriores	
16	Los datos almacenados en la memoria EEPROM de más de ocho bytes se recuperan:	
a	A partir de su lectura directa	
b	Leyendo cada byte	
c	Con un casting	
d	Cualquiera de las anteriores	
e	A+B	
f	B+C	
g	A+C	
h	Ninguna de las anteriores	
17	La máscara combinada con las operaciones bit a bit permite:	
a	Seleccionar parte de los bits de una variable	
b	Detectar el estado de parte de los bits de una variable	
c	Modificar parte de los bits de una variable	
d	Todas las anteriores	
e	A+B	
f	B+C	
g	A+C	
18	¿A qué tipo de dato convierte la función String() un dato numérico?	
a	A su valor decimal	
b	A su valor binario	
c	A su valor hexadecimal	
d	Al tipo de dato que se le indique como parámetro en la función	

19	En el código: String cadena1="10"; String cadena2="9"; cadena1.compareTo(cadena2); la última instrucción devuelve:	
a	Un valor positivo	
b	Un valor negativo	
c	Un valor 0	
d	Da error	
20	En el código: String cadena1="hola"; String cadena2="Hola"; cadena1.equals(cadena2); la última instrucción devuelve:	
a	Un valor positivo	
b	Un valor negativo	
c	Un valor 0	
d	TRUE	
e	FALSE	
f	C ó E	
21	En el código: string cadena1="hola"; string cadena2="Hola"; cadena1.equalsIgnoreCase(cadena2); la última instrucción devuelve:	
a	Un valor positivo	
b	Un valor negativo	
c	Un valor 0	
d	TRUE	
e	FALSE	
f	Da error	

22	En el código: string cadena1="hola"; string cadena2="o"; cadena1.indexOf(cadena2); la última instrucción devuelve:	
a	0	
b	1	
c	2	
d	3	
e	4	
f	-1	
23	En el código: string cadena1="hola"; string cadena2="o"; cadena1.indexOf(cadena2,1); la última instrucción devuelve:	
a	0	
b	1	
c	2	
d	3	
e	4	
f	-1	
24	En el código: string cadena1="hola"; string cadena2="o"; cadena1.indexOf(cadena2,2); la última instrucción devuelve:	
a	0	
b	1	
c	2	
d	3	
e	4	
f	-1	

25	En el código: string cadena1="hola"; string cadena2="a"; cadena1.lastIndexOf(cadena2); la última instrucción devuelve:	
a	0	
b	1	
c	2	
d	3	
e	4	
f	-1	
26	En el código: string cadena1="hola"; string cadena2="a"; cadena1.lastIndexOf(cadena2,2); la última instrucción devuelve:	
a	0	
b	1	
c	2	
d	3	
e	4	
f	-1	
27	En el código: string cadena1="albaricoque"; string cadena2="q"; cadena1.charAt(cadena1.indexOf(cadena2)); la última instrucción devuelve:	
a	a	
b	l	
c	b	
d	r	
e	c	
f	q	

28	En el código: string cadena1="albaricoque"; cadena1.substring(8,9); la última instrucción devuelve:	
a	a	
b	l	
c	b	
d	r	
e	c	
f	q	
29	En el código: string cadena1="albaricoque"; cadena1.substring(8); la última instrucción devuelve:	
a	coque	
b	alba	
c	baco	
d	que	
e	-1	
f	Da error	
30	El código: cadena1="4.25"; float dato=cadena1.toFloat(); dato=dato+0.1111; cadena1=String(dato,5); Serial.println(cadena1); imprime en la consola:	
a	4.36110	
b	4.11110	
c	Da error	

31	El código: cadena1="4,25"; float dato=cadena1.toFloat(); dato=dato+0.1111; cadena1=String(dato,5); Serial.println(cadena1); imprime en la consola:	
a	4.36110	
b	4.11110	
c	Da error	
32	¿Qué método se puede usar para asegurar que se escribe físicamente en la tarjeta cualquier dato pendiente de grabar en un fichero?	
a	Close()	
b	Print()	
c	Write()	
d	Flush()	
e	Todas las anteriores	
f	A+B	
g	B+C	
h	A+D	
33	¿Qué es el escrutinio?	
a	Un procedimiento de recuento automático implementado en Arduino	
b	Un procedimiento estándar incluido en la librería SPI	
c	Un método para acceder a la información de los periféricos	
d	Todas las anteriores	
e	A+B	
f	B+C	
g	A+C	
h	Ninguna de las anteriores	
34	¿Qué es el sondeo?	
a	Un procedimiento de sondeo automático implementado en Arduino	
b	Un procedimiento estándar incluido en la librería SPI	
c	Un método para acceder a la información de los periféricos	
d	Todas las anteriores	
e	A+B	
f	B+C	
g	A+C	
h	Ninguna de las anteriores	

35	El polling:	
a	Es un método para acceder a la información de los periféricos	
b	Es muy eficiente	
c	Consiste en que el propio procesador se encargue de sondear los dispositivos periféricos cada cierto tiempo y de forma continua	
d	Todas las anteriores	
e	A+B	
f	B+C	
g	A+C	
h	Ninguna de las anteriores	
36	¿Es el RESET una interrupción enmascarable?	
a	Sí	
b	No	
37	El índice numérico 'n':	
a	Organiza el acceso a la tabla de vectores de interrupción	
b	Asigna la prioridad de ejecución cuando se producen dos interrupciones de manera simultánea	
c	Todas las anteriores	
d	Ninguna de las anteriores	
38	La prioridad de las interrupciones:	
a	Se asigna mediante un registro especial	
b	Va implícita en el valor del vector de interrupción que le corresponde	
c	Se asigna mediante software	
d	Todas las anteriores	
e	A+B	
f	A+C	
g	B+C	
h	Ninguna de las anteriores	
39	¿Cuál es el bit de habilitación global de interrupciones?	
a	El bit 7 del registro SREG	
b	El bit 0 del registro SREG	
c	El bit 7 del registro EIMSK	
d	El bit 0 del registro EIMSK	
e	Ninguna de las anteriores	

40	La instrucción sei();:	
a	Pone a 0 el bit de máscara de la interrupción	
b	Pone a 1 el bit de máscara de la interrupción	
c	Pone a 0 el bit de habilitación global de interrupciones	
d	Pone a 1 el bit de habilitación global de interrupciones	
e	Ninguna de las anteriores	
41	La instrucción cli();:	
a	Pone a 0 el bit de máscara de la interrupción	
b	Pone a 1 el bit de máscara de la interrupción	
c	Pone a 0 el bit de habilitación global de interrupciones	
d	Pone a 1 el bit de habilitación global de interrupciones	
e	Ninguna de las anteriores	
42	SREG es el:	
a	Registro de habilitación global de interrupciones	
b	Registro de máscaras de interrupciones externas	
c	Registro de control de interrupciones externas	
d	Registro de banderas de interrupciones externas	
e	Ninguna de las anteriores	
43	EIMSK es el:	
a	Registro de habilitación global de interrupciones	
b	Registro de máscaras de interrupciones externas	
c	Registro de control de interrupciones externas	
d	Registro de banderas de interrupciones externas	
e	Ninguna de las anteriores	
44	EICRA es el:	
a	Registro de habilitación global de interrupciones	
b	Registro de máscaras de interrupciones externas	
c	Registro de control de interrupciones externas	
d	Registro de banderas de interrupciones externas	
e	Ninguna de las anteriores	
45	EIFR es el:	
a	Registro de habilitación global de interrupciones	
b	Registro de máscaras de interrupciones externas	
c	Registro de control de interrupciones externas	
d	Registro de banderas de interrupciones externas	
e	Ninguna de las anteriores	

46	PCICR es el:	
a	Registro de control de la interrupción de cambio en el pin	
b	Registro de máscara de interrupción de cambio en el pin	
c	Registro de banderas de interrupción de cambio en el pin	
d	Ninguna de las anteriores	
47	PCMSK es el:	
a	Registro de control de la interrupción de cambio en el pin	
b	Registro de máscara de interrupción de cambio en el pin	
c	Registro de banderas de interrupción de cambio en el pin	
d	Ninguna de las anteriores	
48	PCIFR es el:	
a	Registro de control de la interrupción de cambio en el pin	
b	Registro de máscara de interrupción de cambio en el pin	
c	Registro de banderas de interrupción de cambio en el pin	
d	Ninguna de las anteriores	
49	¿Cómo se puede evitar el rebote en el tratamiento de las interrupciones de Arduino?	
a	Mediante software	
b	Mediante hardware	
c	Todas las anteriores	
d	No se puede evitar	
50	¿Qué hace la función millis()?	
a	Retrasa la ejecución del programa principal el número de milisegundos del valor que se le pasa como parámetro	
b	Retrasa la ejecución del programa principal el número de microsegundos del valor que se le pasa como parámetro	
c	Registra los milisegundos transcurridos desde el inicio del programa	
d	Registra los microsegundos transcurridos desde el inicio del programa	
e	Ninguna de las anteriores	
51	La función millis():	
a	Basa su funcionamiento en el TIMER0	
b	Basa su funcionamiento en el TIMER1	
c	Basa su funcionamiento en el TIMER2	
d	Ninguna de las anteriores	

52	La librería SERVO:	
a	Basa su funcionamiento en el TIMER0	
b	Basa su funcionamiento en el TIMER1	
c	Basa su funcionamiento en el TIMER2	
d	Ninguna de las anteriores	
53	La función tone():	
a	Basa su funcionamiento en el TIMER0	
b	Basa su funcionamiento en el TIMER1	
c	Basa su funcionamiento en el TIMER2	
d	Ninguna de las anteriores	
54	¿Qué hace la función tone()?	
a	Genera una onda senoidal de la frecuencia que se le pase por parámetro	
b	Genera una onda cuadrada de la frecuencia que se le pase por parámetro	
c	Genera una onda triangular de la frecuencia que se le pase por parámetro	
d	Ninguna de las anteriores	
55	¿Se pueden generar interrupciones sin llegar al límite del temporizador?	
a	No, para modificar el tiempo total del temporizador está la función perscaler()	
b	Si, el temporizador mismo genera la interrupción en el momento que nos convenga	
c	Si, a partir de la unidad de comparación	
56	¿Para qué sirve el registro TCNTn?	
a	Para almacenar el conteo	
b	Para almacenar el valor de comparación	
c	Para almacenar información de configuración del temporizador	
d	Todas las anteriores	
e	Ninguna de las anteriores	
57	¿Para qué sirve el registro OCRn?	
a	Para almacenar el conteo	
b	Para almacenar el valor de comparación	
c	Para almacenar información de configuración del temporizador	
d	Todas las anteriores	
e	Ninguna de las anteriores	

58	¿Para qué sirve el registro TCCRn?	
a	Para almacenar el conteo	
b	Para almacenar el valor de comparación	
c	Para almacenar información de configuración del temporizador	
d	Todas las anteriores	
e	Ninguna de las anteriores	
59	El registro TIMSK:	
a	Controla la habilitación/deshabilitación de las interrupciones asociadas al temporizador	
b	Indica las interrupciones asociadas al temporizador que están pendientes de atender	
c	Todas las anteriores	
d	Ninguna de las anteriores	
60	El registro TIMFR:	
a	Controla la habilitación/deshabilitación de las interrupciones asociadas al temporizador	
b	Indica las interrupciones asociadas al temporizador que están pendientes de atender	
c	Todas las anteriores	
d	Ninguna de las anteriores	

RESPUESTAS

1	h		16	f		31	b		46	a
2	a		17	d		32	h		47	b
3	d		18	d		33	c		48	c
4	e		19	b		34	c		49	c
5	c		20	f		35	g		50	c
6	d		21	d		36	b		51	a
7	f		22	b		37	c		52	b
8	f		23	b		38	b		53	c
9	d		24	f		39	a		54	b
10	a		25	d		40	d		55	c
11	e		26	f		41	c		56	a
12	a		27	f		42	a		57	c
13	b		28	f		43	b		58	c
14	f		29	d		44	c		59	a
15	g		30	a		45	d		60	b